Constrained Convex Neyman-Pearson Classification Using an Outer Approximation Splitting Method

Michel Barlaud, Patrick Louis Combettes, Lionel Fillatre

Constrained Convex Neyman-Pearson Classification Using an Outer Approximation Splitting Method

Michel Barlaud, barlaud@I3s.unice.fr
Laboratoire I3S 2000 Route des Lucioles 06903 Sophia Antipolis, France
Patrick L. Combettes, plc@ljll.math.upmc.fr
Sorbonne Universités – UPMC Univ. Paris 06,
UMR 7598, Laboratoire Jacques-Louis Lions, F-75005 Paris, France
Lionel Fillatre, lionel.fillatre@i3s.unice.fr
Laboratoire I3S 2000 Route des Lucioles 06903 Sophia Antipolis, France

June 8, 2015

Abstract

This paper presents an algorithm for Neyman-Pearson classification. While empirical risk minimization approaches focus on minimizing a global risk, the Neyman-Pearson framework minimizes the type II risk under an upper bound constraint on the type I risk. Since the $0/1$ loss function is not convex, optimization methods employ convex surrogates that lead to tractable minimization problems. As shown in recent work, statistical bounds can be derived to quantify the cost of using such surrogates instead of the exact 1/0 loss. However, no specific algorithm has yet been proposed to actually solve the resulting minimization problem numerically. The contribution of this paper is to propose an efficient splitting algorithm to address this issue. Our method alternates a gradient step on the objective surrogate risk and an approximate projection step onto the constraint set, which is implemented by means of an outer approximation subgradient projection algorithm. Experiments on both synthetic data and biological data show the efficiency of the proposed method.

Convex optimization, machine learning, calibrated risk, splitting algorithm

1 Introduction

Support Vector Machine (SVM) is a powerful and well-established method for machine learning [10; 28]. Standard SVM methods use the hinge loss as a convex surrogate to the $0/1$ loss. Generally speaking, the choice of the surrogate loss impacts significantly statistical properties [3]. When using the classical empirical risk minimization approach, the majority class is well classified, whereas the minority class is poorly classified. In many applications, however, the minority class is often the most relevant. For example, in biological applications, patients with pathology are of more interest although they constitute a minority class. Consequently, controlling false negative rates is of utmost importance in biomedical diagnosis.

The Neyman-Pearson is an alternative to the classical empirical risk minimization approach minimizing a global classification risk consisting of a weighted sum of type I and type II risks. The Neyman-Pearson approach minimizes the type II risk subject to an upper bound on the type I risk. To the best of our knowledge, the only previous work dealing with constrained Neyman-Pearson framework was on statistical evaluation [7; 18; 24; 23; 27]. These approaches provide a quantitative relationship between the minimization of the empirical risk and the minimization of the $0/1$ risk. However, they did not propose any algorithm for solving the numerical optimization problem, which consists of minimizing an empirical objective subject to an empirical constraint.

The present paper deals with binary classification. Our main contribution is to propose an implementable algorithm with guaranteed convergence for solving the Neyman-Pearson classification problem. Section 2 deals with state-of-the-art statistical results on Neyman-Pearson classification. Section 3 presents our new splitting algorithm. Finally, Section 4 presents experiments on both synthetic and real RNA-seq lung cancer data from the TGCA dataset.

2 Classification risk and classifiers

2.1 Empirical risk minimization

Henceforth, the $\mathbb{R}^d$-valued random vector X represents a feature vector, the $\{-1,1\}$-valued random variable Y represents the associated label indicating to which class X belongs, and P denotes the underlying probability measure. A classifier is a mapping $h\colon \mathbb{R}^d \mapsto [-1,1]$, the sign of which returns the predicted class given X. An error occurs when $\mathsf{Y}h(X) \leqslant 0$. The classification risk associated with a classifier h is

$$R(h) = \mathsf{P}[\mathsf{Y}h(X) \leqslant 0] = \mathsf{E}\big(1_{]-\infty,0]}(\mathsf{Y}h(X))\big), \tag{1}$$

where $1_{]-\infty,0]}$ denotes the characteristic function of $]-\infty,0]$, i.e., the 0/1 loss function. The above formulation leads in general to numerically intractable optimization problems and it must be simplified. In the present paper, we focus on linear classifiers, meaning that the function h is of the form

$$h_w\colon \mathbb{R}^d \to \mathbb{R}\colon x \mapsto \langle x \mid w\rangle = x^\top w, \tag{2}$$

for some weight vector $w \in \mathbb{R}^d$. In addition, we shall replace the nonconvex 0/1 loss function $1_{]-\infty,0]}$ in (1) by a suitable convex surrogate, i.e., a convex function $\phi\colon \mathbb{R} \mapsto [0,+\infty[$ which approximates $1_{]-\infty,0]}$ (see Fig. 1). This leads to the surrogate risk

$$R_\phi(h) = \mathsf{E}\big(\phi(\mathsf{Y}h(X))\big) = \mathsf{E}\big(\phi(\mathsf{Y}\langle X \mid w\rangle)\big). \tag{3}$$

Our problem set-up is as follows. We assume that m annotated samples $(x_i)_{1\leqslant i\leqslant m}$ in $\mathbb{R}^d$ are available, resulting from the observation of independent realizations of the feature vector X. The associated realizations $(\mathsf{y}_i)_{1\leqslant i\leqslant m}$ of the label Y are variables valued in $\{-1,+1\}$, which represent the two classes. The features $(x_i)_{1\leqslant i\leqslant m}$ are problem-dependent, e.g., bag of word and Fisher vectors in image classification, gene expression in biological application, etc. In the classical empirical risk minimization approach, the goal is to learn w by minimizing the surrogate empirical risk

$$\varphi\colon \mathbb{R}^d \to \mathbb{R}\colon w \mapsto \frac{1}{m}\sum_{i=1}^{m} \phi\big(\mathsf{y}_i\langle x_i \mid w\rangle\big). \tag{4}$$

In the context of empirical risk minimization, Bartlett et al. [3] provide a general quantitative relationship between the risk using the $0/1$ loss and the risk using a surrogate loss function $\phi\colon \mathbb{R} \to \mathbb{R}$. They show that this relationship gives upper bounds on the excess risk under the provision that the convex loss ϕ is calibrated, i.e., ϕ is differentiable at 0 with $\phi'(0) < 0$.

2.2 Controlling false alarms

The type I risk associated with a classifier h on $\mathbb{R}^d$ is

$$R^-(h) = \mathsf{P}(\mathsf{Y}h(X) \leqslant 0 \,|\, \mathsf{Y} = -1), \tag{5}$$

while the type II risk is defined by

$$R^+(h) = \mathsf{P}(\mathsf{Y}h(X) \leqslant 0 \,|\, \mathsf{Y} = +1). \tag{6}$$

The Neyman-Pearson classification consists in solving

$$\underset{R^-(h)\leqslant\alpha}{\text{minimize}} \;\; R^+(h), \tag{7}$$

where $\alpha \in \,]0,1[$ is user-defined. Now, let us introduce the ϕ-type I risk and the ϕ-type II risk associated with a classifier h as

$$R^-_\phi(h) = \mathsf{E}(\phi(\mathsf{Y}h(X)) \,|\, \mathsf{Y} = -1) \tag{8}$$

and

$$R^+_\phi(h) = \mathsf{E}(\phi(\mathsf{Y}h(X)) \,|\, \mathsf{Y} = +1), \tag{9}$$

respectively. Let P^{Y} denote the conditional distribution of X given Y. We split the set of samples $(x_i)_{1\leqslant i\leqslant m}$ into the subset $(x_i^-)_{1\leqslant i\leqslant m^-}$ of features which have distribution P^{-1}, and the complementary subset $(x_i^+)_{1\leqslant i\leqslant m^+}$ of features which have distribution P^{+1} (the sample sizes m^- and m^+ are deterministic). The empirical surrogate ϕ-risks of a classifier h are defined by

$$\widehat{R}^-_\phi(h) = \frac{1}{m^-}\sum_{i=1}^{m^-} \phi(-h(x_i^-)) \quad \text{and} \quad \widehat{R}^+_\phi(h) = \frac{1}{m^+}\sum_{i=1}^{m^+} \phi(h(x_i^+)). \tag{10}$$

2.3 Constrained Neyman-Pearson framework

To process unbalanced datasets, classical methods rely on the weighted objective

$$\widehat{R}_{\phi,\rho} = \widehat{R}^-_\phi + \rho\widehat{R}^+_\phi, \tag{11}$$

where ρ controls the unbalanced dataset. An alternative is a cost-sensitive approach with Lagrangian formulation such as C-SVM or ν-SVM [12; 29], where the parameter ν controls the trade-off between the two types of errors. Unfortunately, Lagrangian methods require cross validation for tuning the parameter ρ or ν. It is well known that this cross validation can lead to poor classification accuracy. To circumvent this issue, we propose the following constrained Neyman-Pearson calibrated classification approach. The Neyman-Pearson paradigm is an alternative to the empirical risk minimization

approach in statistical learning. It attempts to find a classifier $\tilde{h}$ as a solution to the constrained optimization problem

$$\underset{\widehat{R}_\phi^-(h)\leqslant\tau}{\text{minimize}}\ \widehat{R}_\phi^+(h), \tag{12}$$

where $\tau \in]0,1[$ is chosen as a function of α in such a way that the relaxed constraint $\widehat{R}_\phi^-(\tilde{h}) \leqslant \tau$ on the type I risk implies that the constraint $R^-(\tilde{h}) \leqslant \alpha$ on the type I risk in (5) is satisfied by $\tilde{h}$ with high probability. The statistical properties of $\tilde{h}$ are given in the following theorem initially established in [18] for aggregate classifiers and applied here to bounded linear classifiers. A bounded linear classifier is a special case of an aggregate classifier.

Let $\alpha \in]0,1[$, let $\delta \in]0,1/2[$, let $\mathcal{H}$ be the set of linear classifiers $h_w\colon x \mapsto \langle x \mid w\rangle$ such that $\|w\|_1 \leqslant M$ for some fixed $M \in]0,+\infty[$, and set

$$\mathcal{H}_\phi^\alpha = \big\{h \in \mathcal{H} \;\big|\; R_\phi^-(h) \leqslant \alpha\big\}. \tag{13}$$

Set

$$\varepsilon = \inf\big\{\theta \in]0,1[\;\big|\; \mathcal{H}_\phi^{\theta\alpha} \neq \varnothing\big\} \tag{14}$$

and let $\phi\colon \mathbb{R} \to \mathbb{R}$ be a differentiable convex function with a β-Lipschitz derivative. Assume that there exists $\theta \in]0,1[$ such that $\mathcal{H}_\phi^{\theta\alpha} \neq \varnothing$. Set

$$\gamma = 4\beta\sqrt{2\log\left(\frac{4d}{\delta}\right)} \tag{15}$$

and

$$m_0 = \inf\left\{m \in \mathbb{N} \;\middle|\; m \geqslant \left(\frac{4\gamma}{\alpha(1-\varepsilon)}\right)^2\right\}, \tag{16}$$

and suppose that there exists $\xi \in \mathbb{R}$ such that $\|X\|_\infty \leqslant \xi$ almost surely. Assume that $m^- \geqslant m_0$ and let $\tilde{h}$ be a solution to (12), where $\tau = \alpha - \gamma/\sqrt{m^-}$. Then, with probability at least $1-2\delta$,

$$R^-(\tilde{h}) \leqslant R_\phi^-(\tilde{h}) \leqslant \alpha \tag{17}$$

and

$$R_\phi^+(\tilde{h}) - \inf_{h\in\mathcal{H}_\phi^\alpha} R_\phi^+(h) \leqslant \frac{4\gamma\,\phi(1)}{(1-\varepsilon)\alpha\sqrt{m^-}} + \frac{2\gamma}{\sqrt{m^+}}. \tag{18}$$

This theorem shows that the classifier $\tilde{h}$ obtained by minimizing the empirical surrogate ϕ-risk is a solution to (7). It also shows that the ϕ-type II risk of $\tilde{h}$ tends to the minimum ϕ-type II risk over $\mathcal{H}_\phi^\alpha$ as $\min\{m^-, m^+\}$ becomes arbitrarily large. The convergence rate of the ϕ-type II risk is bounded by (18).

2.4 Smooth calibrated loss

We restrict our attention to calibrated convex surrogate losses that satisfy the following properties [6].

Assumption 1 *The loss $\phi\colon \mathbb{R} \to \mathbb{R}$ is convex, everywhere differentiable with a Lipschitz-continuous gradient, and twice differentiable at 0 with $\phi''(0) = \max \phi''$. Furthermore, there exists an increasing function $f\colon \mathbb{R} \to [0,1]$ which is antisymmetric with respect to the point $(0, f(0)) = (0, 1/2)$ such that $(\forall t \in \mathbb{R})\ \phi(t) = -t + \int_{-\infty}^{t} f(s)ds$.*

This interesting class of smooth calibrated losses [11], allows us to compute the posterior estimation without Platt estimation [17]. The function f maps directly a real-valued prediction h of a sample x_i to a posterior estimation

$$\mathsf{P}[\mathsf{Y}_i = +1 | x_i] = f(h(x_i)) \tag{19}$$

for the class $+1$. Now, in connection with (4), consider the convex function

$$\varphi^+\colon \mathbb{R}^d \to \mathbb{R}\colon w \mapsto \frac{1}{m^+} \sum_{i=1}^{m^+} \phi\big(\langle x_i^+ \mid w \rangle\big). \tag{20}$$

Then, under Assumption 1, φ^+ is differentiable and its gradient

$$\nabla\varphi^+\colon w \mapsto \frac{1}{m^+} \sum_{i=1}^{m^+} \phi'\big(\langle x_i^+ \mid w \rangle\big) x_i^+, \tag{21}$$

has Lipschitz constant $1/\beta$, where

$$\beta = \frac{m^+}{\phi''(0) \sum_{i=1}^{m^+} \|x_i^+\|^2}. \tag{22}$$

Let us note that computer vision classification involves normalized high dimensional features such as Fisher vectors [21]. In this case, (22) reduces to

$$\beta = \frac{1}{\phi''(0)}. \tag{23}$$

Examples of functions which satisfy Assumption 1 include that induced by $f\colon t \mapsto 1/(1+\exp(-t))$, which leads to the logistic loss

$$\phi\colon t \mapsto \ln(1 + \exp(-t)), \tag{24}$$

for which $\phi''(0) = 0.25$. Other examples are the calibrated version of the linear hinge loss

$$\phi\colon t \mapsto \max\{0, -t\} - \ln(\sqrt{2 + |t|}) + \ln(2), \tag{25}$$

as well as the Matsusita loss [15]

$$\phi\colon t \mapsto \frac{1}{2}(-t + \sqrt{1 + t^2}). \tag{26}$$

Note that the boosting exponential loss does not satisfy the above properties, and that neither does the hinge loss $\phi\colon t \mapsto \max\{1, -t\}$ used in classical SVM.

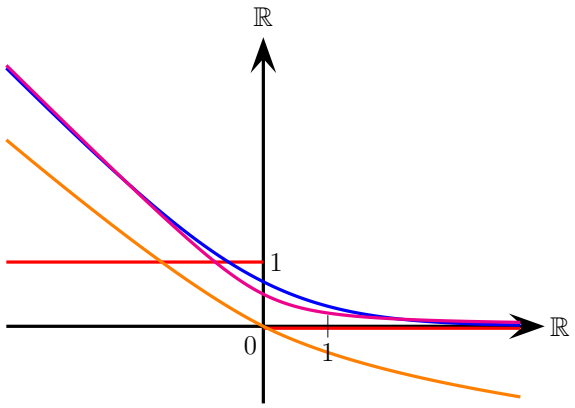

Figure 1: Convex surrogate functions for the 0/1 loss function $1_{]-\infty,0]}$ (in red): the calibrated hinge loss $\phi\colon t \mapsto \max(0,-t) + \ln(2) - \ln(\sqrt{2+|t|})$ (in orange), the logistic loss $\phi\colon t \mapsto \ln(1+e^{-t})$ (in blue), and the Matsusita loss $\phi\colon t \mapsto (\sqrt{t^2+1}-t)/2$ (in magenta).

3 Splitting algorithm

3.1 General framework

In this section, we propose an algorithm for solving the Neyman-Pearson classification problem (12). This algorithm fits in the general category of forward-backward splitting methods, which have been popular since their introduction in signal processing and machine learning [9; 16; 22; 25]. These methods offer flexible implementation with guaranteed convergence of the sequence of iterates they generate, a key property to ensure the reliability of our variational classification scheme.

The minimization problem (12) can be recast as follows.

Problem 1 *Suppose that ϕ satisfies Assumption 1, define φ^+ as in (20), define*

$$\varphi^-\colon \mathbb{R}^d \to \mathbb{R}\colon w \mapsto \frac{1}{m^-}\sum_{i=1}^{m^-} \phi\big(\mathsf{y}_i\langle x_i^- \mid w\rangle\big), \tag{27}$$

and set

$$C = \Big\{ w \in \mathbb{R}^d \;\Big|\; \varphi^-(w) \leqslant \tau \Big\}. \tag{28}$$

The problem is to

$$\underset{w\in C}{\text{minimize}} \;\; \varphi^+(w). \tag{29}$$

Let us note that for reasonably sized data set, φ^+ or φ^- will be coercive and hence the Problem 1 will have at least one solution [5, Proposition 11.14]. As noted in Section 2.4, φ^+ is a differentiable convex function and its gradient has Lipschitz constant $1/\beta$, where β is given by (22). Likewise, since φ^- is convex and continuous, C is a closed convex set. The principle of a splitting method is to use the constituents of the problems, here φ^+ and C, separately [5]. In the problem at hand, it is natural to use the projection-gradient method to solve (29). This method, which is an instance of the proximal forward-backward algorithm [5], alternates a gradient step on the objective φ^+ and a projection step onto the constraint set C. Given $w_0 \in \mathbb{R}^d$, a sequence $(\gamma_n)_{n\in\mathbb{N}}$ of strictly positive parameters, and a sequence $(a_n)_{n\in\mathbb{N}}$ in $\mathbb{R}^d$ modeling computational errors in the implementation of the projection operator P_C, it assumes the form

$$\begin{array}{l} \text{for } n = 0, 1, \dots \\ \left\lfloor \begin{array}{l} v_n = w_n - \gamma_n \nabla \varphi^+(w_n) \\ w_{n+1} = P_C v_n + a_n. \end{array} \right. \end{array} \tag{30}$$

In view of (21), (30) can be rewritten as

$$\begin{array}{l} \text{for } n = 0, 1, \dots \\ \left\lfloor \begin{array}{l} v_n = w_n - \dfrac{\gamma_n}{m^+} \displaystyle\sum_{i=1}^{m^+} \phi'\big(\mathsf{y}_i \langle x_i^+ \mid w \rangle\big) \mathsf{y}_i x_i^+ \\ w_{n+1} = P_C v_n + a_n. \end{array} \right. \end{array} \tag{31}$$

We derive at once from [9, Theorem 3.4(i)] the following convergence result, which guarantees the convergence of the iterates.

Let $w_0 \in \mathbb{R}^d$, let $(\gamma_n)_{n\in\mathbb{N}}$ be a sequence in $]0, +\infty[$, and let $(a_n)_{n\in\mathbb{R}^d}$ be a sequence in $\mathbb{R}^d$ such that

$$\sum_{n\in\mathbb{N}} \|a_n\| < +\infty, \quad \inf_{n\in\mathbb{N}} \gamma_n > 0, \quad \text{and} \quad \sup_{n\in\mathbb{N}} \gamma_n < 2\beta. \tag{32}$$

Then the sequence $(w_n)_{n\in\mathbb{N}}$ generated by (31) converges to a solution to Problem 1.

The implementation of (31) is straightforward except for the computation of $P_C v_n$. Indeed, C is defined in (27) as the lower level set of a convex function, and no explicit formula exists for computing the projection onto such a set. Fortunately, Proposition 3.1 asserts that $P_C v_n$ does not have to be computed exactly. We can therefore implement it approximately by performing a sufficient number of iterations of an efficient algorithm designed for computing the projection onto a lower level set of a convex function. Next, we describe such an algorithm, which is borrowed from [4] and proceeds by successive outer approximations generated by subgradient projections.

3.2 Projection algorithm

[4, Corollary 6.10(ii)] Let $p_0 \in \mathbb{R}^d$, let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a differentiable convex function, and let

$\eta \in \mathbb{R}$ be such that $D = \big\{p \in \mathbb{R}^d \;\big|\; f(p) \leqslant \eta\big\} \neq \varnothing$. Iterate

$$
\begin{array}{l}
\text{for } k = 0, 1, \ldots \\
\quad \text{if } f(p_k) \leqslant \eta \\
\quad\quad \lfloor \text{terminate.} \\
\quad p_{k+1/2} = p_k + \dfrac{\eta - f(p_k)}{\|\nabla f(p_k)\|^2} \nabla f(p_k) \\
\quad \chi_k = \big\langle p_0 - p_k \;\big|\; p_k - p_{k+1/2} \big\rangle \\
\quad \mu_k = \|p_0 - p_k\|^2 \\
\quad \nu_k = \|p_k - p_{k+1/2}\|^2 \\
\quad \rho_k = \mu_k \nu_k - \chi_k^2 \\
\quad \text{if } \rho_k = 0 \text{ and } \chi_k \geqslant 0 \\
\quad\quad \lfloor p_{k+1} = p_{k+1/2} \\
\quad \text{if } \rho_k > 0 \text{ and } \chi_k \nu_k \geqslant \rho_k \\
\quad\quad \left\lfloor p_{k+1} = p_0 + \left(1 + \dfrac{\chi_k}{\nu_k}\right)\big(p_{k+1/2} - p_k\big) \right. \\
\quad \text{if } \rho_k > 0 \text{ and } \chi_k \nu_k < \rho_k \\
\quad\quad \left\lfloor \begin{array}{l} p_{k+1} = p_k + \dfrac{\nu_k}{\rho_k}\Big(\chi_k\big(p_0 - p_k\big) \\ \qquad\qquad\qquad + \mu_k\big(p_{k+1/2} - p_k\big)\Big) \end{array} \right.
\end{array}
\tag{33}
$$

Then either the algorithm terminates in a finite number of iterations at $P_D p_0$ or it generates an infinite sequence $(p_k)_{k\in\mathbb{N}}$ such that $p_k \to P_D p_0$.

The principle of the above algorithm is as follows (see Fig. 2). At iteration k, if $f(p_k) \leqslant \eta$, then $p_k \in D$ and the algorithm terminates with $p_k = P_D p_0$. Otherwise, one first computes the subgradient projection

$$
p_{k+1/2} = p_k + \frac{\eta - f(p_k)}{\|\nabla f(p_k)\|^2} \nabla f(p_k) \tag{34}
$$

of p_k onto D. We have [4]

$$
D \subset H(p_k, p_{k+1/2}) = \big\{p \in \mathbb{R}^d \;\big|\; \big\langle p - p_{k+1/2} \;\big|\; p_k - p_{k+1/2} \big\rangle \leqslant 0\big\}. \tag{35}
$$

The closed half-space $H(p_0, p_{k+1/2})$ serves as an outer approximation to D at iteration k. On the other hand, by construction, we also have a second similar outer approximation, namely

$$
D \subset H(p_0, p_k) = \big\{p \in \mathbb{R}^d \;\big|\; \langle p - p_k \;|\; p_0 - p_k \rangle \leqslant 0\big\}. \tag{36}
$$

As a result, $D \subset H(p_0, p_k) \cap H(p_k, p_{k+1/2})$. The update p_{k+1} is computed as the projection of p_0 onto the outer approximation $H(p_0, p_k) \cap H(p_k, p_{k+1/2})$. This update can be expressed explicitly in terms of the vectors $p_k - p_0$ and $p_{k+1/2} - p_k$ as described above. This provides a convergent sequence the limit of which is the projection of p_0 onto D.

3.3 Practical implementation

We have observed that (33) yields in a few iterations to a point close to the exact projection of p_0 onto D. This can be measured by the magnitude of the gap $f(p_k) - \eta$ since $p_k = P_D p_0 \Leftrightarrow f(p_k) \leqslant \eta$.

We can therefore insert the subroutine (33) into (31) (with $p_0 = v_n$, $f = \varphi^-$, and $\eta = \tau$) to evaluate approximately $P_C v_n$ by performing only K_n iterations of it at iteration n. In this case, it follows from (33) that (31) reduces to

$$
\begin{array}{l}
\text{for } n = 0, 1, \dots \\
\quad v_n = w_n - \dfrac{\gamma_n}{m^+} \displaystyle\sum_{i=1}^{m^+} \phi'\big(\mathsf{y}_i \langle x_i^+ \mid w_n \rangle\big) \mathsf{y}_i x_i^+ \\
\quad p_0 = v_n \\
\quad \text{for } k = 0, 1, \dots, K_n \\
\qquad \eta_k = \alpha - \dfrac{1}{m^-} \displaystyle\sum_{i=1}^{m^-} \phi\big(\mathsf{y}_i \langle x_i^- \mid p_k \rangle\big) \\
\qquad \text{if } \eta_k \geqslant 0 \\
\qquad\quad \text{terminate.} \\
\qquad u_k = \dfrac{1}{m^-} \displaystyle\sum_{i=1}^{m^-} \phi'\big(\mathsf{y}_i \langle x_i^- \mid p_k \rangle\big) \mathsf{y}_i x_i^- \\
\qquad p_{k+1/2} = p_k + \dfrac{\eta_k}{\|u_k\|^2} u_k \\
\qquad \chi_k = \langle p_0 - p_k \mid p_k - p_{k+1/2} \rangle \\
\qquad \mu_k = \|p_0 - p_k\|^2 \\
\qquad \nu_k = \|p_k - p_{k+1/2}\|^2 \\
\qquad \rho_k = \mu_k \nu_k - \chi_k^2 \\
\qquad \text{if } \rho_k = 0 \text{ and } \chi_k \geqslant 0 \\
\qquad\quad p_{k+1} = p_{k+1/2} \\
\qquad \text{if } \rho_k > 0 \text{ and } \chi_k \nu_k \geqslant \rho_k \\
\qquad\quad p_{k+1} = p_0 + \left(1 + \dfrac{\chi_k}{\nu_k}\right)(p_{k+1/2} - p_k) \\
\qquad \text{if } \rho_k > 0 \text{ and } \chi_k \nu_k < \rho_k \\
\qquad\quad p_{k+1} = p_k + \dfrac{\nu_k}{\rho_k}\Big(\chi_k (p_0 - p_k) \\
\qquad\qquad\qquad + \mu_k \big(p_{k+1/2} - p_k\big)\Big) \\
\quad w_{n+1} = p_{K_n}.
\end{array}
\tag{37}
$$

3.4 Convergence of the inner loop

The theory shows that we need perform only K_n inner iterations as long as can guarantee that the approximation errors $(\|a_n\|)_{n \in \mathbb{N}}$ form a summable sequence. Consider iteration k of (37). Then, since $D \subset H(p_0, p_k)$ and p_k is the projection of p_0 onto $H(p_0, p_k)$, we have $\|p_k - P_D p_0\| \leqslant \|p_0 - P_D p_0\|$. Hence $p_k \in D \Leftrightarrow p_k = P_D p_0$, i.e., $f(p_k) \leqslant \eta \Leftrightarrow p_k = P_D p_0$. Now suppose that, for every k, $f(p_k) > \eta$ (otherwise we are done). By convexity, f is Lipschitz-continuous on compact sets [5, Corollary 8.32], and therefore there exists a constant ζ such that $0 < f(p_k) - \eta = f(p_k) - f(P_D p_0) \leqslant \zeta \|p_k - P_D p_0\| \to 0$. In addition, since in our case $\operatorname{int} D \neq \varnothing$, using standard error bounds on convex inequalities [20], there exists a constant ξ such that $\|p_k - P_D p_k\| \leqslant \xi f(p_k) - \eta$. So we can approximate the order of the error $\|a_n\|$ by that of $f(p_{K_n}) - \eta$, which is readily computable. In practice, however, we have found

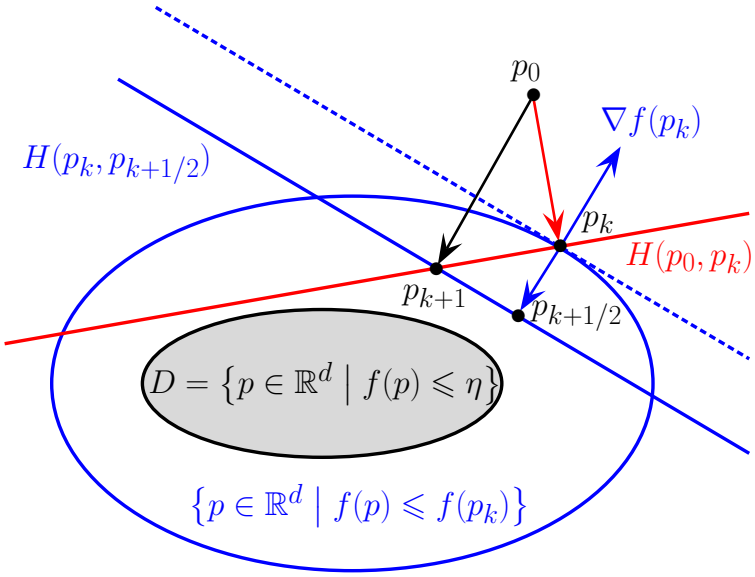

Figure 2: A generic iteration of (33) for computing the projection of p_0 onto D. At iteration k, the current iterate is p_k and D is contained in the half-space $H(p_0, p_k)$ onto which p_k is the projection of p_0 (see (36)). If $f(p_k) > \eta$, the gradient vector $\nabla f(p_k)$ is normal to the lower level set $\{p \in \mathbb{R}^d \mid f(p) \leqslant f(p_k)\}$, and the subgradient projection $p_{k+1/2}$ of p_k onto D is defined by (34); it is the projection of p_k onto the half-space $H(p_k, p_{k+1/2})$ of (35), which contains D. The update p_{k+1} is the projection of p_0 onto $H(p_0, p_k) \cap H(p_k, p_{k+1/2})$.

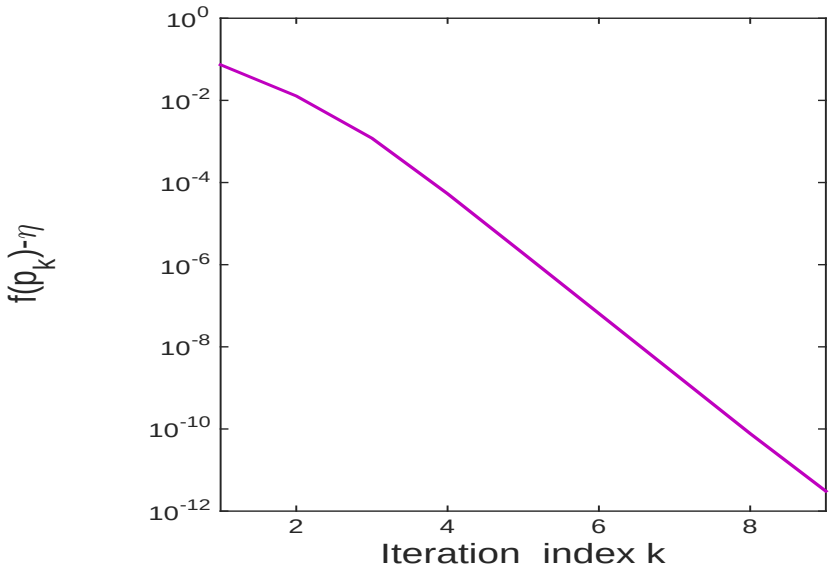

Figure 3: Convergence of the projection inner loop.

such an analysis to be superfluous as the inner loop converges extremely fast and typically (Fig. 3) $f(p_k) - \eta \leqslant 10^{-12}$ after just $k = 9$ iterations. So we have taken $K_n = 9$ in our experiments.

4 Experimental evaluation

4.1 Setting

Classification and risk prediction based on gene transcription factor and clinical data sets in cancer analysis is currently a challenging task. Since the early classification work of [13; 14] using DNA microarray data sets, state of the art classification methods have been based on empirical risk minimization approaches such as support vector machines; see the recent review [26] on feature selection for classification for more details. To the best of our knowledge no alternate optimization algorithm has been proposed for solving the constrained Neyman-Pearson classification Problem 1. For this reason, we cannot perform comparisons in this experimental study.

In all experiments we use the logistic loss (24) as a surrogate. We use half of the data for training and half for testing, and we set $\alpha = 0.1$. In the figures, we plot the $0/1$ risk and surrogate risks or the classical global and mean accuracy as a function of the number of iterations of (37). We show in Fig. 4 a typical convergence pattern of (37).

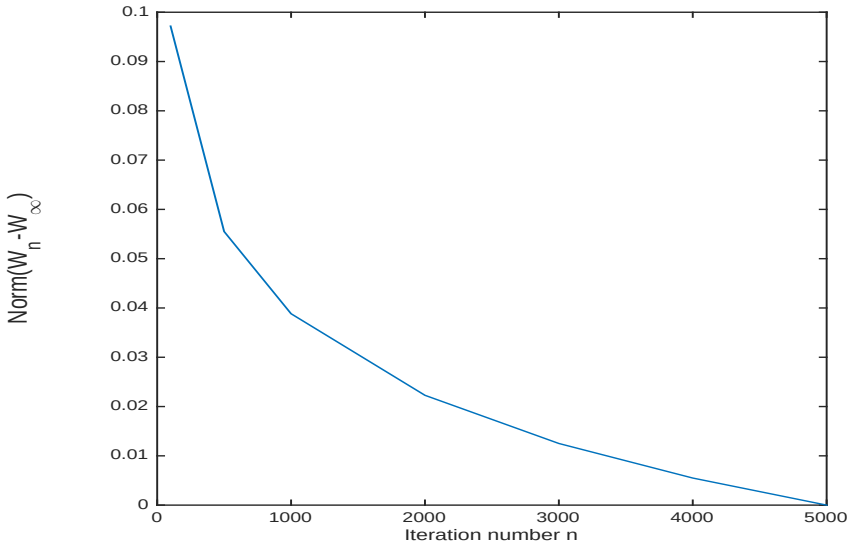

Figure 4: A typical convergence pattern for (37).

4.2 RNA-seq model and preprocessing

RNAseq is a recent high-throughput sequencing technology.[1] The distribution model of RNA-seq is different from DNA microarray data and requires adapted preprocessing. The underlying distribution model of RNAseq is a negative binomial distribution [19]. Let X_{ij} denote the observed raw read count for gene i and library j, where $1 \leqslant i \leqslant p$ and $1 \leqslant j \leqslant$ the count X_{ij} follows a negative binomial distribution where m_{ij} is the mean and ψ_i is the dispersion for gene i. The mean satisfies

$$m_{ij} = \mu_i \, L_i \, D_j, \tag{38}$$

where L_i is the length of gene i, D_j is proportional to the total number of reads for library j (also called the sequencing depth), and μ_i is the true and unknown expression level for gene i. We propose to use a simple transformation, known to have an optimum property (i.e., to be the best of that degree of complexity) for m_{ij} large and $\psi_i \geqslant 1$ (see details in [2])

$$Z_{ij} = \ln\left(X_{ij} + \frac{1}{2}\psi_i\right). \tag{39}$$

The transformation (39) makes the distribution of Z_{ij} closer to a monovariate normal distribution. Its variance is approximately $\Psi'(\psi_i)$, where $\Psi'(t)$ denotes the second derivative of $\ln\Gamma(t)$ with respect to t. The mean of Z_{ij} is approximately given by [2]

$$\mathsf{E}(Z_{ij}) \quad \approx \quad \ln\mu_i + \ln L_i + \ln D_j - \frac{1}{2\psi_i} + \ln\left(1 + \frac{\psi_i}{2m_{ij}}\right). \tag{40}$$

The last term in (40) is negligible when $\psi_i \ll m_{ij}$.

[1] The first commercially available RNA sequencer (454 Life Sciences Pyrosequencer) was marketed in 2005.

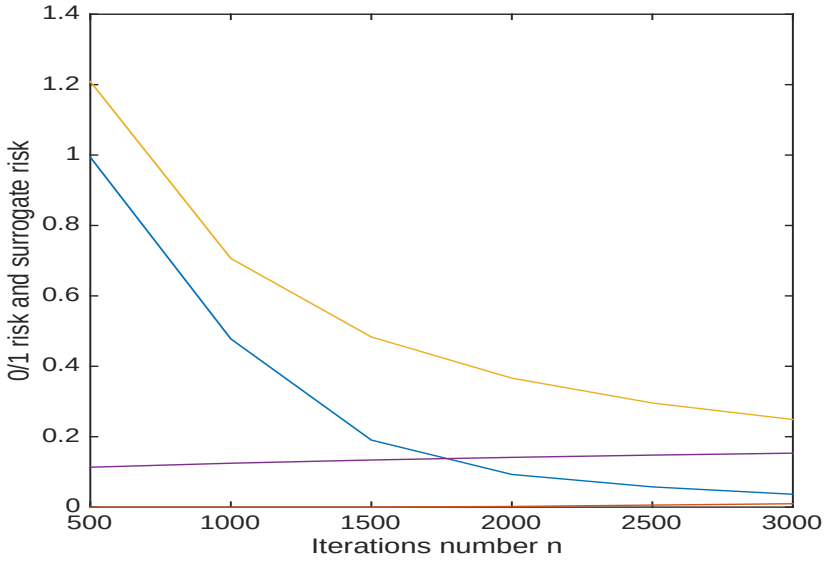

Figure 5: Synthetic data, training, $0/1$ risk type I (in orange) and type I surrogate risk (in magenta) $0/1$ risk type II (in blue) and type II surrogate risk (in yellow).

4.3 Results on a synthetic dataset

We have generated artificial negative binomial samples for the counts X_{ij} with 1000 genes for each patient. We have 3340 patients in the first class and only 1040 patients in the minority class. The length L_i of each gene is known and $\psi_i = 6$ for each gene i. The sequencing depths D_j are generated as realizations of a Gaussian variable modelling the experimental variability. For the first class, the μ_i's are chosen arbitrarily. The choice is based on typical values estimated from real RNAseq measurements. For the second class, 20% of the μ_i's (randomly chosen) of the first class are changed: their values are increased or decreased randomly, by using Gaussian distributed offsets. The random nature of the D_j's has no impact on classification. Finally, the counts X_{ij} are generated by using a negative binomial random generator. We then applied the transformation (39) to obtain the observations Z_{ij}.

The challenge is to predict whether an artificial patient belongs to one class or the other. The data set is unbalanced since we have 3340 samples in one class and only 1040 samples in the minority class consider as type I. We display in Fig. 5 the performance of the algorithm in the training set. In this experiment, the projection algorithm requires only $K_n = 9$ iterations (which is typical). In addition, the surrogate type I risk (in magenta) is perfectly controlled and induces a type I risk (in orange) close to zero with the inequalities

$$R^-(\tilde{h}) \leqslant R^-_\phi(\tilde{h}) \leqslant \alpha. \tag{41}$$

This means that the statistical upper bound from [18] is overdetermined. Furthermore the estimated type II risk (in blue) is less than the type II surrogate risk (in yellow) as

$$R^-(\tilde{h}) \leqslant R^+_\phi(\tilde{h}). \tag{42}$$

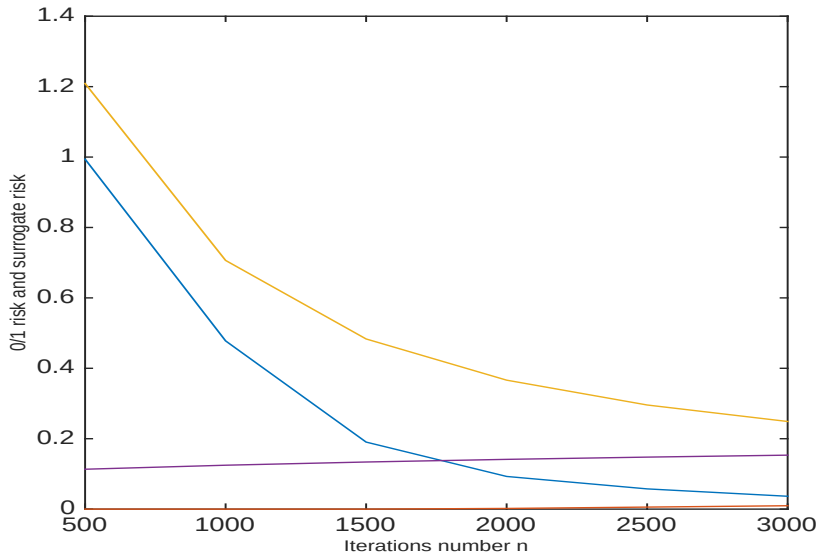

Figure 6: Synthetic data, testing, $0/1$ risk type I (in orange) and type I surrogate risk (in magenta), $0/1$ risk type II (in blue) and type II surrogate risk (in yellow).

We have similar results in the test set. Fig. 6 shows the performance of the algorithm in the test set. It can be noted that the constraint on the type I surrogate risk (in magenta) induces a type I risk (in orange) close to zero.

Fig. 7 shows the classical accuracy evaluation: mean accuracy (blue), and global accuracy (red).

4.4 Results on the lung cancer RNAseq TCGA dataset

In this real experiment, we use the lung cancer RNAseq data set from the TGCA dataset (The Cancer Genome Atlas) [1]. The data set is highly unbalanced since we have 452 tumoral samples and only 58 samples without tumor. The goal is to predict from the RNAseq dataset whether there is a tumor or not. We use a classical filtering method for a coarse gene selection [13], [14], where the score

$$S_i = \frac{|\mu_i^+ - \mu_i^-|}{\sigma_i^+ + \sigma_i^-} \tag{43}$$

is close to the Fisher score, and μ_i is the mean and σ_i the standard deviation in each class. Fig. 8 shows the performance of the algorithm in the test set. Similarly to experiment on synthetic data set, It can be noted that the constraint on the type I surrogate risk (in magenta) induces a type I risk (in orange) close to zero.

Fig. 9 shows the performance of the algorithm for the test set. The constraint on the type I surrogate risk (in magenta), induces a type I risk (in orange) close to zero. The estimated type II risk (in blue) is smaller than the type II surrogate risk (in yellow).

Fig. 10 shows the mean accuracy (blue) and the global accuracy (red) in the test set.

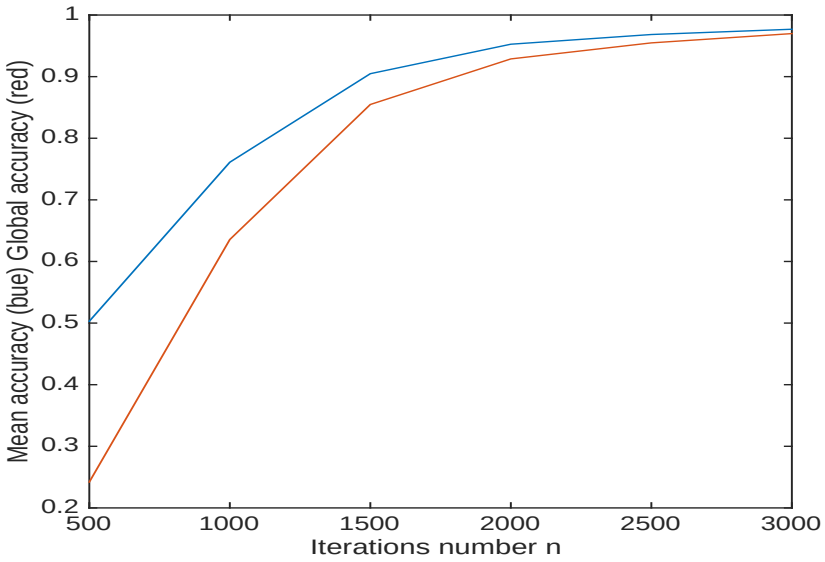

Figure 7: Synthetic data, testing, global (in blue) and mean (in red) accuracy.

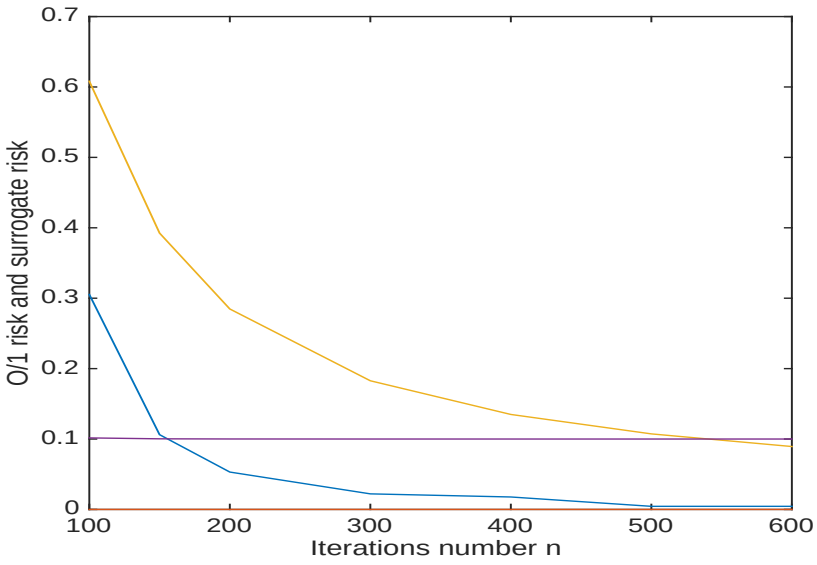

Figure 8: Tumor, training, $0/1$ risk type I (in orange) and type I surrogate risk (in magenta) $0/1$ risk type II (in blue) and type II surrogate risk (in yellow).

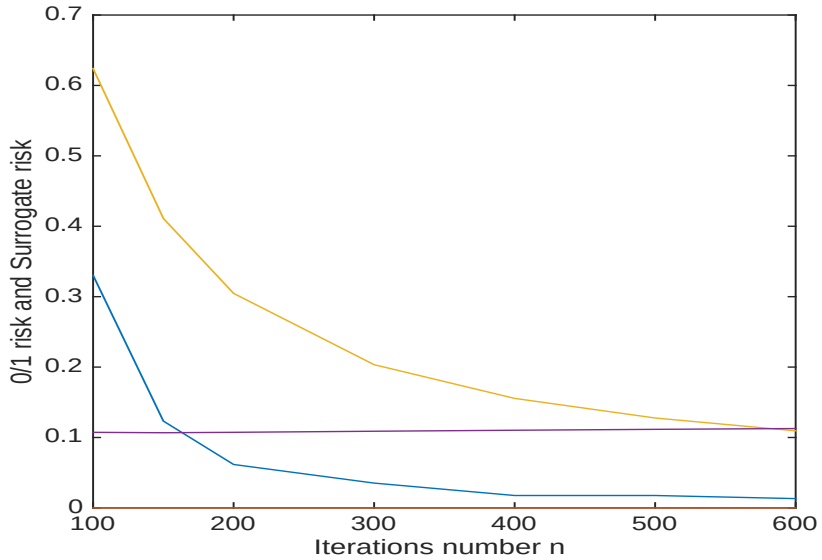

Figure 9: Tumor, Testing, $0/1$ risk type I (in orange) and type I surrogate risk (in magenta) $0/1$ risk type II (in blue) and type II surrogate risk (in yellow).

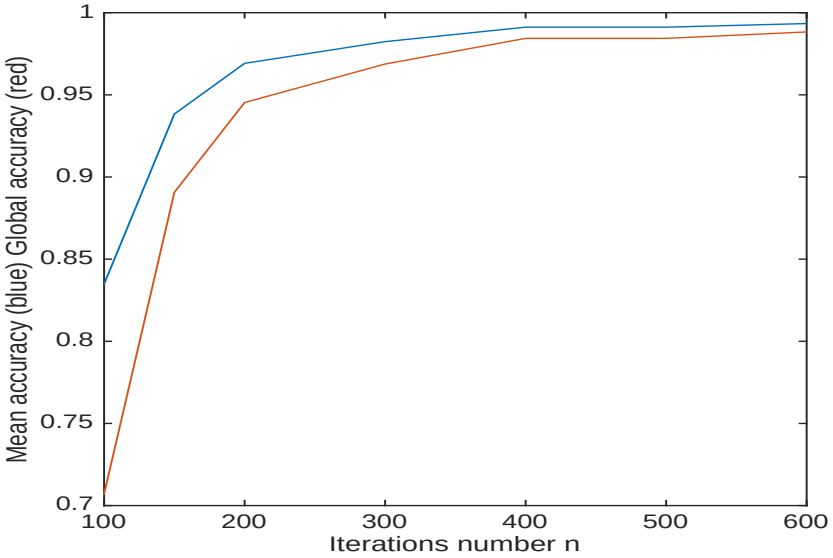

Figure 10: Tumor, global (in blue) and mean (in red) accuracy.

An extensive study on lung cancer data set using our new algorithm is currently being evaluated by colleagues in biology.

5 Conclusion and future work

We have proposed an efficient algorithm to solve the Neyman-Pearson classification problem. Assuming that the surrogate loss is smooth with a Lipschitz gradient, we have provided a new algorithm which alternates a gradient step on the objective surrogate loss and an approximate projection step onto the constraint set. Let us note that we have presented Proposition 3.2 (and, therefore, algorithm (37)) with a single constraint. However, the results of [4; 8] allow for the use of several constraints (each is then activated by its own subgradient projector). Thus, additional information about the problem can be easily be injected in (29), in particular in the form of constraints on w. This will be explored elsewhere. Experiments on both synthetic data and biological data show the efficiency of our new method. On-going work includes joint feature selection (such as DNA mutations) and classification for the Neyman-Pearson classification problem.

6 Appendix: Proof of Theorem 2.3

Set $\mathcal{B}_M = \big\{w \in \mathbb{R}^d \;\big|\; \|w\|_1 \leqslant M\big\}$ and let $\{e_i\}_{1\leqslant i\leqslant d}$ be the canonical basis of $\mathbb{R}^d$. Then $\mathcal{B}_M$ is the convex hull of the set

$$S = \{s_i\}_{1\leqslant i\leqslant 2d} = \{\pm Me_i\}_{1\leqslant i\leqslant d}, \tag{44}$$

where $s_i = Me_i$ for $1 \leqslant i \leqslant d$ and $s_i = -Me_{i-d}$ for $d+1 \leqslant i \leqslant 2d$. Hence, for every $w \in \mathcal{B}_M$, there exists $(\lambda_i^w)_{1\leqslant i\leqslant 2d} \in [0,+\infty[^{2d}$ such that $\sum_{i=1}^{2d} \lambda_i^w = 1$ and

$$w = \sum_{i=1}^{2d} \lambda_i^w s_i. \tag{45}$$

It follows that the associated bounded linear classifier h_w is

$$h_w \colon x \mapsto \langle x \mid w\rangle = \sum_{i=1}^{2d} \lambda_i^w \langle x \mid s_i\rangle. \tag{46}$$

Let us note that

$$\big(\forall x = (x_i)_{1\leqslant i\leqslant d} \in \mathbb{R}^d\big) \quad \langle x \mid s_i\rangle = \begin{cases} Mx_i & \text{if } \; i \in \{1,\ldots,d\},\\ -Mx_{i-d} & \text{if } \; i \in \{d+1,\ldots,2d\}. \end{cases} \tag{47}$$

Under the assumption that $\|X\|_\infty \leqslant \xi$ a.s., it follows that, without any loss of generality, the feature vector X can be normalized so that $\|X\|_\infty \leqslant 1$ a.s. Hence $\langle \cdot \mid s_i\rangle \colon \mathbb{R}^d \mapsto [-1,1]$. As shown in [18, Theorem 5], a classifier which is a convex combination of base classifiers from $\mathbb{R}^d$ to $[-1,1]$ satisfies the properties given in Theorem 2.3. According to (46), a bounded linear classifier h_w is a convex combination of the base classifiers $(\langle \cdot \mid s_i\rangle)_{1\leqslant i\leqslant 2d}$ and hence a bounded linear classifier satisfies the required properties. The conclusion therefore follows from [18, Theorem 5].

References

[1] https://tcga-data.nci.nih.gov/tcga/tcgaHome2.jsp

[2] F. J. Anscombe. The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 35:246–254, 1948.

[3] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101:138–156, 2006.

[4] H. H. Bauschke and P. L. Combettes. A weak-to-strong convergence principle for Fejér-monotone methods in Hilbert spaces. *Mathematics of Operations Research*, 26:248–264, 2001.

[5] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York, 2011.

[6] W. BelHajAli, R . Nock, and M. Barlaud. Minimizing calibrated loss using stochastic low-rank newton descent for large scale image classification. In *International Conference on Pattern Recognition*, 2014.

[7] A. Cannon, J. Howse, D. Hush, and C. Scovel. Learning with the Neyman-Pearson and min-max criteria. Technical report, Los Alamos National Laboratory, 2002.

[8] P. L. Combettes. Strong convergence of block-iterative outer approximation methods for convex optimization. *SIAM Journal on Control and Optimization*, 38:538–565, 2000.

[9] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4:1168–1200, 2005.

[10] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, New York, 2000.

[11] R. D'Ambrosio, R. Nock, W. Bel Haj Ali, F. Nielsen, and M. Barlaud. Boosting nearest neighbors for the efficient estimation of posteriors. In *European Conference on Machine Learning '12*, 2012.

[12] M. A. Davenport, R. G. Baraniuk, and C. Scott. Tuning support vector machines for minimax and Neyman-Pearson classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1888–1898, 2010.

[13] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16:906–914, 2000.

[14] I. Guyon, J. Weston, S. Barnhill, W. Vapnik, and N. Cristianini. Gene selection for cancer classification using support vector machines. In *Machine Learning*, pp. 389–422, 2002.

[15] K Matsusita. Distance and decision rules. *Annals of the Institute of Statistical Mathematics*, 16:305–315, 1964.

[16] S. Mosci, L. Rosasco, S. Matteo, A. Verri, and S. Villa. Solving structured sparsity regularization with proximal methods. In *Machine Learning and Knowledge Discovery in Databases*, vol. 6322, pp. 418–433, 2010.

[17] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press, 1999.

[18] P. Rigollet and X. Tong. Neyman-Pearson classification, convexity and stochastic constraints. *Journal of Machine Learning Research*, 12:2831–2855, 2011.

[19] M. D. Robinson and A. Oshlack. A scaling normalization method for differential expression analysis of rna-seq data. *Genome Biology*, 11:R25, 2010.

[20] S. M. Robinson. An application of error bounds for convex programming in a linear space. *SIAM Journal on Control*, 13:271–273, 1975.

[21] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the Fisher vector: Theory and practice. *International Journal of Computer Vision*, 105:222–245, 2013.

[22] M. Schmidt, N. L. Roux, and F. R. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in Neural Information Processing Systems 24*, pp. 1458–1466. 2011.

[23] C. Scott. Performance measures for Neyman-Pearson classification. *IEEE Transactions on Information Theory*, 53:2852–2863, 2007.

[24] C. Scott and R. Nowak. A Neyman-Pearson approach to statistical learning. *IEEE Transactions on Information Theory*, 51:3806–3819, 2005.

[25] S. Sra, S. Nowozin, and S. J. Wright (eds.). *Optimization for Machine Learning*. MIT Press, Cambridge, MA, 2011.

[26] J. Tang, S. Alelyani, and H. Liu. Feature selection for classification: A review. *Data Classification: Algorithms and Applications.* C. Aggarwal (ed.), CRC Press, 2014.

[27] X. Tong. A plug-in approach to Neyman-Pearson classification. *Journal of Machine Learning Research*, 14:3011–3040, 2013.

[28] V. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.

[29] K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on AI*, pages 55–60, 1999.